Greedy Search-Binary PSO Hybrid for Biclustering Gene Expression Data

Shyama Das
Department of Computer Science
Cochin University of Science and
Technology, Kochi, Kerala, India

Sumam Mary Idicula
Department of Computer Science,
Cochin University of Science and
Technology, Kochi, Kerala, India,

ABSTRACT

As a useful data mining technique biclustering identifies local patterns from gene expression data. A bicluster of a gene expression dataset is a subset of genes which exhibit similar expression patterns along a subset of conditions. In this paper a new method is introduced based on greedy search algorithm combined with the evolutionary technique particle swarm optimization for the identification of biclusters. Greedy methods have the possibility of getting trapped in local minima. Metaheuristic methods like particle swarm optimization have features for escaping from local minima and can find global optimal solutions. In this algorithm biclusters are identified in three steps. In the first step small disjoint tightly coregulated submatrices are generated using K-Means clustering algorithm. Then greedy search algorithm is used to enlarge the seeds. Output of greedy search algorithm is used as initial population of binary PSO. The result obtained on Yeast dataset shows that this method can generate high quality biclusters.

Categories and Subject Descriptors

D.3.3 [**Programming Languages**]: Language Constructs and Features – *abstract data types, polymorphism, control structures.* This is just an example, please use the correct category and subject descriptors for your submission.

General Terms

Your general terms must be any term which can be used for general classification of the submitted material such as Pattern Recognition, Security, Algorithms et. al.

Keywords

Biclustering, gene expression data, greedy search, kmeans clustering, particle swarm optimization.

1. INTRODUCTION

DNA Microarray technology measures the expression level of thousands of genes under different conditions simultaneously in a single experiment. The applications of microarray technology range from gene functional annotation, the usefulness in the medical domain for aid in more accurate diagnosis, prognosis, treatment planning, drug discovery and protein network analysis. The data generated from this technology is a high dimensional matrix where rows represent genes and columns represent experimental conditions or samples. The experimental conditions can be patients, tissue types, different time points etc. The gene expression datasets typically contain thousands of genes and hundreds of conditions. Each element in the matrix refers to the expression level of a particular gene under a specific condition. Each entry in this matrix is a real number. Genes participating in the same biological process will have similar expression patterns. Clustering is the suitable mining method for identifying these patterns.

Being high dimensional data mining algorithms for finding various patterns from microarray gene expression data has immense application in bioinformatics and clinical research. One of the most widely used data mining technique for the analysis of gene expression data is clustering. Clustering groups similar genes or conditions. Clustering of co-expressed gene into biologically meaningful groups helps in inferring the biological role or function of new gene that is co-expressed with a known gene.

However clustering based on the entire row has many disadvantages and restrictions in implementation process. Clustering is based on the assumption that all the related genes behave similarly across all the measured conditions. It may reveal the genes which are very closely co-regulated. Based on a general understanding of the cellular process, the subsets of genes are co-regulated and co-expressed under certain experimental conditions. But they behave almost independently under other conditions. Moreover clustering happens to partition the genes into disjoint sets i.e. each gene is associated with a single biological function, which in fact is in contradiction to the biological system [1].

To overcome the problems of clustering concept of biclustering was introduced. Biclustering was first introduced by Hartigan and called it direct clustering [2]. Biclustering is clustering applied along the row and column, simultaneously. Clustering is a global model where as biclustering is a local model. Biclustering approach identifies the genes which show similar expression levels under a specific subset of experimental conditions. In biclustering the objective is to identify maximal subgroups of genes and subgroups of conditions such that the genes express highly correlated activities over a range of conditions. Cheng and Church were the first to apply biclustering to gene expression data [3]. Biclustering is also known as coclustering, bidimensional clustering and subspace clustering. The application of biclustering is ideal when some genes have multiple functions and experimental conditions are different.

2. MATERIALS AND METHODS

2.1 Model of Bicluster

Gene expression dataset is a matrix in which rows represent genes and columns represent experimental conditions. An element aij of the expression matrix A represents the logarithm of the relative abundance of the mRNA of the ith gene under the jth condition. A bicluster of a gene expression dataset is a subset of gene which exhibit similar expression patterns along a subset of conditions. The rows and columns of the bicluster need not be contiguous as in the expression matrix.

There are different types of biclusters [1]. A bicluster with coherent values identifies a subset of genes and a subset of conditions with coherent values on both rows and columns. In order to measure the degree of coherence a measure called mean squared residue score or hscore was introduced by Cheng and Church. It is the sum of the squared residue score. The residue score of an element aij in a submatrix Aij is defined as RS(i,j)=*aij-aIj-aiJ+aIJ*

Hence mean squared residue score or

$$MSR(I,J) = \frac{1}{|I||J|} \sum i \in I, j \in J \; (aij - aIj - aiJ + aIJ)^2$$

Where I denotes the row set, J denotes the column set, aij denotes the element in a submatrix, aiJ denotes the ith row mean, aIj denotes the jth column mean, and aIJ denotes the mean of the whole bicluster.

A submatrix A_{ij} is called a δ bicluster if MSR(I,J)< δ for some δ >0. A high MSR value signifies that the data is uncorrelated. A low MSR value means that there is correlation in the matrix. The value of δ depends on the dataset. For yeast dataset the value of δ is 300.

2.2 Encoding of Biclusters

Each bicluster is encoded as a binary string [4]. The length of the string is the number of rows plus the number of columns of the gene expression data matrix .A bit is set to one when the corresponding gene or condition is included in the bicluster. This representation is advantageous for node addition and deletion.

2.3 Seed Finding

In this algorithm a simple seed finding technique is used [5].

Using the kmeans algorithm the gene expression dataset is partitioned into n gene clusters and m sample clusters. In order to get maximum 10 genes per gene cluster, it is further divided according to the cosine angle distance from the cluster centre. Similarly each sample cluster is further divided into sets of 5 samples according to cosine angle distance from the cluster centre. Suppose that the number of gene clusters, having maximum 10 close genes is p and number of sample clusters having maximum 5 conditions is q. The initial gene expression data matrix is thus partitioned into p*q submatrices and bicluster seeds having hscore value below a certain limit is selected to initialize the bicluster.

2.4 Greedy Search Algorithm

In the seed growing phase each seed is enlarged separately by adding more genes and conditions. A separate list is maintained for conditions and genes not included in the seed bicluster. First conditions are added followed by genes. In modified greedy search algorithm the best element is selected from the gene list or condition list and added to the bicluster. The quality of the element is determined by the hscore or MSR value of the bicluster after including the element in the bicluster. The element which results in minimum hscore value when added to the bicluster is considered as the best element. It cannot be specified as an element with smallest incremental cost of hscore because adding some elements reduces the hscore value. Seed growing starts from condition list followed by gene list until the hscore value reaches the given threshold. This is a greedy method since our aim is to select the next element which produces bicluster with minimum hscore value [6] [7].

2.5 Initial Population for PSO

PSO is a population based evolutionary optimization algorithm. Usually PSO is initialized with a population of random solutions. In this study the results obtained from greedy search algorithm is used to initialize PSO. This will result in faster convergence compared to random initialization. Maintaining diversity in the population is another advantage of initializing with biclusters from greedy search method. Moreover greedy methods suffer from local minima problem which can be eliminated by methods like PSO.

2.6 PSO Based Biclustering

The particle swarm optimization is proposed by Kennedy and Eberhart [8] while attempting to simulate the choreographed, graceful motion of swarms of birds trying to find food. It is a heuristics based optimization approach. Particle swarm optimization (PSO) is a population based evolutionary computation algorithm. The members of the whole population are maintained throughout the search procedure. The potential solution of PSO is named as particles and each one is assigned a randomized velocity. Each particle is flown to the optimal solution in the solution space. PSO does not use the filtering operation such as crossover and/or mutation used in evolutionary type methods. Convergence speed and relative simplicity are the two important features which makes it suitable for solving the optimization problems. Biclustering as an optimization problem with the objective of finding biclusters with low mean squared residue and high volume PSO is extremely suitable for solving it[10]. Usually PSO is initialized with a population of random solutions. Here the seeds obtained from greedy search are used to initialize PSO.

Each particle of PSO explores a possible solution. It adjusts its flight according to its own and its companions flying experience. The personal best position is the best solution found by the particle during the course of flight. This is denoted by pbest (personal best). The optimal solution attained by the entire swarm is gBest (global best). PSO iteratively updates the velocity of each particle towards its pBest and gBest positions efficiently. For finding an optimal or near-optimal solution to the problem, PSO keeps updating the current generation of particles. Each particle is a candidate for the solution of the problem. The whole function is accomplished by using the information about the best solution obtained by each particle and the entire population. Each particle has got a set of attributes such as current velocity, current position, the best position discovered by the particle so far and, the best position discovered by the entire particle so far. Each particle begins with an initial velocity and position. Thereafter a swarm particle-i will update its own speed and in accordance with the following equations:

$$V\ (i+1)=w*Vi+\ \{Cp*r1*(pBesti - Xi)\}\ +\{Cg*r2*(gBest-Xi)\} \quad (1)$$

$$X(i+1)=Xi+V(i+1) \quad (2)$$

In equation (1), *w* is the inertia weight; r1 and r2 are random numbers within the range {0,1}. Cp is the Cognitive learning rate and Cg is the Social learning rate. gBest is the best particle found so far and pBesti is the best position discovered so far by the corresponding particle.

In binary PSO [9], Vi, is a probability, and it must be constrained to the interval [0, 1]. A logistic transformation S(Vi) is used to convert the value to this range. The consequent change in the position is defined by the following rule: If(rand() < S(Vi)) then Xi = 1;else Xi = 0. The function S(v) is a sigmoid limiting transformation and rand() is a random number selected from a uniform distribution in [0,1].

2.7 Pseudo-code Description for PSO

```
Algorithm psobicluster(seeds,δ,noofparticles, maxiter)
For i=1 to noofparticles
Initialize particle i using seed i generated by greedy search
Initialize velocity of particle i
End (for)
While iterno<=maxiter
For each particle
 Calculate fitness value
If the fitness value is better than the best fitness value (pBest )
set current value as the new pBest
End(if)
End (for)
Select the particle with the best fitness value of all the particles as the gBest
For each particle
Calculate particle velocity according equation (1)
Update particle position according equation (2)
End (for)
End (while)
```

2.8 Fitness Function

The main objective is to find maximal biclusters with low mean squared residue. Given the value of δ (δ>0), the following fitness function can be used to assess the quality of bicluster [10].

G(B(I,J)= |I|.|J| if MSR(I,J) less than or equal to δ

=δ/MSR(i,j) otherwise

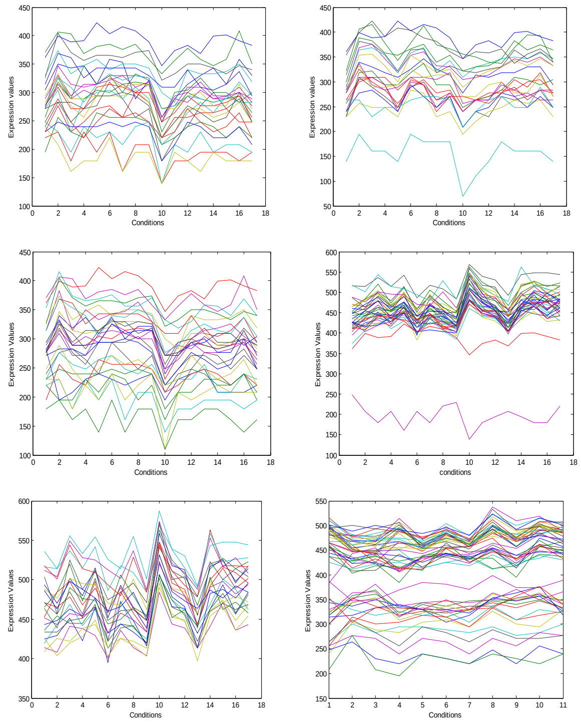
Algorithm is used to maximize the objective function. If the particles in PSO are initialized randomly or if the seeds from K_Means are used to initialize PSO then the result is biclusters with maximum 10 conditions. When greedy search algorithm is used for initialization of particles almost all biclusters contain 17 conditions.

3. EXPERIMENTAL RESULTS

3.1 Datasets used

The proposed algorithm is implemented in Matlab and experiments are conducted on the Yeast Saccharomyces cerevisiae cell cycle expression dataset to evaluate the quality of the proposed method. The dataset is based on Tavazoie et al [11]. Dataset consists of 2884 genes and 17 conditions. The values in the expression dataset are integers in the range 0 to 600. There are 34 missing values represented by -1. The dataset is obtained from http://arep.med.harvard.edu/biclustering.

3.2 Bicluster Plots



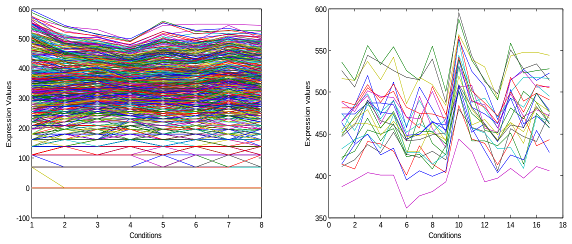


Figure 1. Eight biclusters obtained from the Yeast expression data. Bicluster labels are (a),(b),(c),(d),(e),(f),(g) and (h) respectively. In the bicluster plots X axis contains conditions and Y axis contains expression values. The details about the biclusters can be obtained from Table 1 using bicluster label.

Table 1. Information of biclusters obtained using greedy search -binary PSO algorithm

(a)	25	17	425	195.9666
(b)	21	17	357	178.1294
(c)	28	17	476	189.1636
(d)	36	17	612	195.7957
(e)	22	17	374	146.7061
(f)	54	11	594	192.1012
(g)	500	8	4000	199.4028
(h)	23	17	391	150.2494

4. COMPARISON

Table 2 lists a comparison of results of various algorithms on Yeast data. Performance of greedy search- Binary PSO hybrid with that of SEBI [12], Cheng and Church's algorithm (CC) [3], and the algorithm FLOC by Yang et al. [13], DBF [14] and Modified greedy [6] are given. Here biclusters with MSR less than 100 obtained from greedy search is used as initial population of PSO. Computation time required is very less compared to greedy search running completely to attain the desired MSR. The Avg.MSR for modified greedy binary PSO hybrid is better than all other algorithms except DBF. Average condition number is better than all other algorithms except SEBI. Average gene number is better than SEBI. Largest Bicluster size is same as DBF and better than FLOC and SEBI. Average gene number is better than SEBI.

Table.2 Performance comparison between Greedy Search Binary PSO Hybrid and other algorithms

Algorithm	Avg. MSR	Avg. Volume	Avg. Gene Num.	Avg Cond Num	Largest bicluster size
GS Binary PSO	180.94	903.63	88.62	15.13	4000
DBF	114.70	1627.20	188.00	11.00	4000
SEBI	205.18	209.92	13.61	15.25	1394
Cheng-Church	204.29	1576.98	166.71	12.09	4485
FLOC	187.54	1825.78	195.00	12.80	2000
Modified greedy	185.86	4690.36	515.57	13.36	12645

5. CONCLUSION

In this paper a new algorithm is introduced based on greedy search and Binary PSO, for finding biclusters in gene expression data. In the first step K-Means algorithm is used to cluster rows and columns of the data matrix separately and they are combined to form small tightly co-regulated submatrices. More genes and conditions are added to these seeds till the mean squared residue score is less than 100 using a greedy method. The result obtained from greedy search is used for initializing the particles of PSO. The algorithm is implemented on the Yeast Saccharomyces cerevisiae cell cycle expression dataset. In terms of average condition number and average MSR it is better than some of the most popular biclustering algorithms such as the ones listed in Table 1.

6. REFERENCES

[1] Madeira S. C.and Oliveira A. L., "Biclustering algorithms for Biological Data analysis: a survey" IEEE Transactions on computational biology and bioinformatics, pp. 24-45, 2004.

[2] J.A. Hartigan, "Direct clustering of Data Matrix", Journal of the American Statistical Association, 67 (337) pp. 123-129, 1972.

[3] Yizong Cheng and George M. Church, "Biclustering of Gene expression data", Proc Int Conf Intell Syst MolBiol, vol. 8, pp. 93-103, 2000.

[4] Anupam Chakraborty and Hitashyam Maka "Biclustering of Gene Expression Data Using Genetic Algorithm" Proceedings of Computation Intelligence in Bioinformatics and Computational Biology CIBCB, pp 1-8, 2005.

[5] Chakraborty Aand Maka H, "Biclustering of gene expression data by simulated annealing", HPCASIA '05, pp 627-632, 2005.

[6] Shyama Das and Sumam Mary Idicula "Modified Greedy Search algorithm for Biclustering Gene Expression Data" Proceedings of the Int. Conf. ADCOM 2009.

[7] Shyama Das and Sumam Mary Idicula "K-Means Greedy Search Hybrid algorithm for Biclustering Gene Expression

Data” Advances in Computational Biology, Research Book, Springer 2010.

[8] Kennedy J., R. Eberhart, “Particle Swarm Optimization,” Proc. Of IEEE international Conference on Neural Networks (ICW), Australia, 4, pp. 1942-1948, 1995.

[9] Kennedy J. and Eberhart R.C., “A Discrete Binary Version of the Particle Swarm Optimization”, Proc. of the conference on Systems, Man, and Cybernetics SMC97, pp.4104-4109, 1997.

[10] Baiyi Xie, Shihong Chen, Feng Liu, “Biclustering of Gene Expression data using PSO-GA hybrid”, Proc. of the First International Conference on Bioinformatics and Biomedical Engineering, pp.302-305, 2007.

[11] Tavazoie S, Hughes JD, Campbell MJ, Cho RJ and Church GM., “Systematic determination of genetic network architecture”, Nat Genet., vol.22, no.3 pp. 281-285, 1999.

[12] Federico Divina and Jesus S. Aguilar-Ruize, “Biclustering of Expression Data with Evolutionary computation”, IEEE Transactions on Knowledge and Data Engineering, Vol. 18, pp. 590-602, 2006.

[13] J Yang, H Wang, W Wang and P Yu, “Enhanced Biclustering on Expression Data”, Proc. Third IEEE Symp. BioInformatics and BioEng. (BIBE’03), pp. 321-327, 2003.

[14] Z. Zhang, A. Teo, B.C. Ooi, K.L. Tan, Mining deterministic biclusters in gene expression data, in: Proceedings of the fourth IEEE Symposium on Bioinformatics and Bioengineering (BIBE’04), pp. 283-292, 2004.